Hierarchical Image-Region Labeling via Structured Learning

Julian McAuley[1,2]
julian.mcauley@nicta.com.au

Teofilo de Campos[1,3]
t.decampos@st-annes.oxon.org

Gabriela Csurka[1]
gabriela.csurka@xrce.xerox.com

Florent Perronnin[1]
florent.perronnin@xrce.xerox.com

[1] Xerox Research Centre Europe
Meylan, France

[2] Australian National University/NICTA
Canberra, Australia

[3] University of Surrey
Guildford, United Kingdom

(all authors were at Xerox at the time of this work)

Abstract

We present a graphical model which encodes a series of hierarchical constraints for classifying image regions at multiple scales. We show that inference in this model can be performed efficiently and exactly, rendering it amenable to structured learning. Rather than using feature vectors derived from images themselves, our model is parametrised using the outputs of a series of first-order classifiers. Thus our model learns which classifiers are useful at different scales, and also the relationships between classifiers at different scales. We present promising results on the VOC2007 and VOC2008 datasets.

1 Introduction

When classifying and segmenting images, some categories may be possible to identify based on global properties of the image, whereas others will depend on highly local information; many approaches deal with this problem by extracting features at multiple scales. However, segmentation based on local information can be highly noisy unless smoothness constraints are enforced. These two facts present a problem from a learning perspective: while it is possible to learn a first-order classifier (i.e., a classifier based on *local* information) which incorporates information from multiple scales, learning a classifier which enforces smoothness constraints is an example of *structured learning*, which appears to have made very little progress in this area due to the NP-hardness of many smoothness-enforcing algorithms.

In this paper, we will present a tree-structured graphical model that can be used to enforce smoothness constraints, while incorporating image features extracted at multiple scales. The tractability of this model will render it easily amenable to structured learning, which we believe is novel in this kind of segmentation scenario. We will define a loss based on approximate segmentations of an image (such as bounding boxes), allowing us to exploit the large amount of information in datasets such as VOC2007 and VOC2008 [7, 8]. We will use visual features from [6], which appear to exhibit state-of-the-art performance in classification problems, and show that their patch-level classification performance can be improved by structured learning.

BMVC 2009 doi:10.5244/C.23.49

The main contributions in our paper are as follows: firstly, in contrast to many patch-level segmentation schemes, inference in our model is efficient and exact, allowing us to circumvent the problems encountered when performing structured learning with approximate algorithms. Secondly, instead of parametrising our model using image features directly, our model is parametrised using the outputs of a series of first-order classifiers, and can therefore easily extend and combine existing first-order classification approaches.

2 Related literature

The idea of partitioning an image into regions at multiple scales is certainly not novel [11, 16]. However, these papers are solely concerned with global categorisation, whereas we also consider the problem of local region labeling. Furthermore, we shall not use the features of the image regions themselves, but rather our features are probability scores generated by existing first-order models, such as those from [6].

The approach of using tree-structured graphical models to incorporate global data into local segmentation problems (or simply to circumvent the problems associated with grid-structured models) is also not new; see for example [12]. However, to our knowledge, our approach is the first to apply structured-learning in this scenario, in order to improve the classification results of first-order classifiers. Similarly, others use information at an image-level to guide segmentation at the patch-level [23]. Other papers using similar approaches (though for different applications) include [9, 20, 25].

Many papers use lattice-structured Markov Random Fields (MRFs) to deal with the problem of patch-level segmentation. The unary and pairwise terms in such models may be similar to ours, i.e., the pairwise energy typically denotes a smoothness constraint. Energy minimization in such a model is NP-hard in the general case, so approximate forms of inference must be used [30]. There are many examples in this framework, though some important papers include [4] (*α-expansion, $\alpha\beta$-swap*), [22] (*normalised cuts*), and [17] (*log-cut*). [26] presents a comparative study of these ideas. We avoid such models as the need to perform approximate inference is of major concern from a learning perspective.[1] Other recent papers which apply learning to the problem of image segmentation/localisation include [2, 27].

Other authors also use grid structured models, but restrict their potential functions such that the energy minimization problem can be solved exactly. Examples include boolean-submodular functions [3, 15], *lattice*-submodular functions [14], and convex functions [13]. However, the energy functions we wish to use certainly do not fall into any of these categories.

3 Our model

The nodes ($\mathscr{X}$) in our graphical model ($\mathscr{M}$) are similar to the regions used in [11, 16] (though in those papers, they do not form a graph); these nodes are depicted in Figure 1, and will be indexed by $x_{l,(i,j)}$ where l is the node's level in the graphical model, and (i,j) is its grid position. In fact, a similar graphical model has been suggested for binary segmentation in [18].

[1] Some papers make an exception to this rule, such as [5] and [21]. Recently, some theoretical results have been obtained regarding the performance of structured learning in such cases [10].

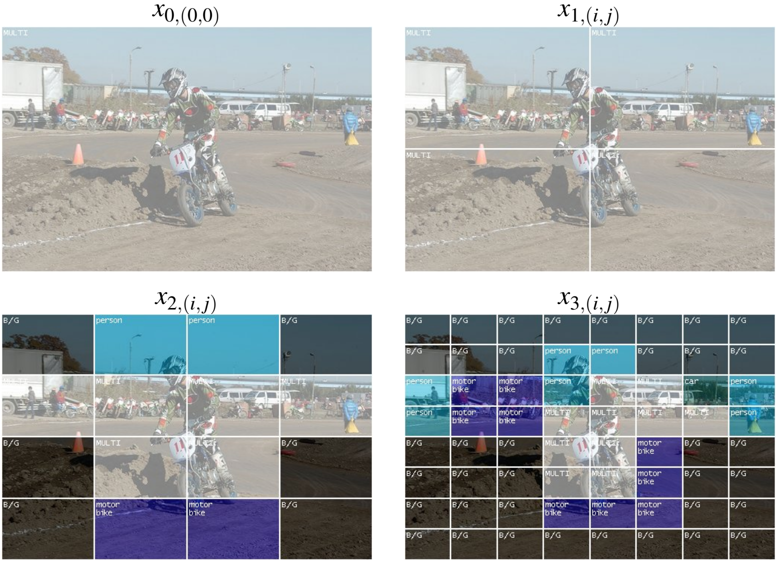

$x_{0,(0,0)}$
MULTI
$x_{1,(i,j)}$
MULTI MULTI
MULTI MULTI
$x_{2,(i,j)}$
B/G person person B/G
MULTI MULTI MULTI MULTI
B/G MULTI MULTI B/G
B/G motor bike motor bike B/G
$x_{3,(i,j)}$
B/G B/G B/G B/G B/G B/G B/G B/G
B/G B/G B/G person person B/G B/G B/G
person motor bike motor bike person MULTI MULTI car person
person motor bike motor bike MULTI MULTI MULTI MULTI person
B/G B/G B/G MULTI MULTI motor bike B/G B/G
B/G B/G B/G MULTI MULTI motor bike B/G B/G
B/G B/G B/G motor bike motor bike motor bike B/G B/G
B/G B/G B/G B/G B/G B/G B/G B/G

Figure 1: Nodes on the first four levels of our model. Nodes are indexed by $x_{l,(i,j)}$ where l is the node's level in the graphical model, and (i,j) is its grid position.

In words, a node on level k is connected to a node on level $k+1$, if and only if the regions corresponding to these nodes overlap. More formally,

$$x_{k,(i,j)} \text{ is connected to } \left\{x_{k+1,(2i,2j)}, x_{k+1,(2i,2j+1)}, x_{k+1,(2i+1,2j)}, x_{k+1,(2i+1,2j+1)}\right\} \tag{1}$$

(equivalently, $x_{k,(i,j)}$ is connected to $x_{k-1,(i/2,j/2)}$). Note that the graphical model is *undirected*. It is worth noting that there are no connections *between* nodes at a given level, and as such we are not enforcing neighbourhood constraints *per se*. Neighbourhood constraints will only be enforced indirectly, due to the fact that neighbors are connected via their parents.[2] Such a formulation is preferable because it results in a *tractable* graphical model (it forms a quadtree), whereas enforcing neighbourhood constraints directly typically requires that we resort to approximate forms of belief propagation.

The number of levels in this graph will depend on the size of the images in question, as well as how densely image patches are sampled. In practice, our graph is built to the maximal depth such that every region contains at least one patch, meaning that the 'regions' on the bottom level are essentially 'patches'. Details are given in Section 5.

3.1 Modeling hierarchical constraints in $\mathscr{M}$

When used in a classification scenario, this model will ensure that nodes on higher levels (i.e., nodes with smaller values of l) will always be assigned to at least as generic a class as nodes on lower levels; this is precisely analogous to the notion of *inheritance* in object-oriented programming. The simplest inheritance diagram (denoted $\mathscr{H}$) that we may use is depicted in Figure 2.

[2] In other words, we assume that two neighbors are *conditionally independent*, given their parents.

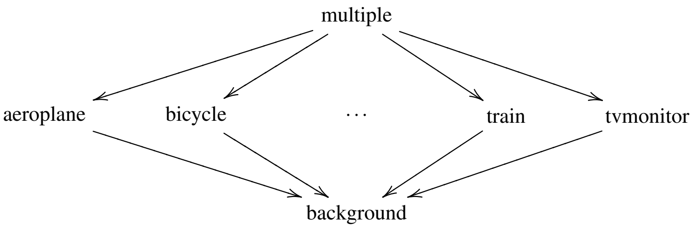


Figure 2: The simplest possible hierarchy; the most general possible assignment simply states that an image region may contain multiple classes; the least general states that a region does not contain any class (↓ denotes 'more general than'). Class labels are taken from [7, 8], though any set of labels could be used.

Note that in $\mathscr{H}$, the class 'background' is actually a child of all specific classes; this is done because our training data may only approximately segment the image (i.e., using a bounding box). Thus an object that is classified as 'cat' on a higher level may be separated into 'cat' and 'background' on the level below. In principle, we could add additional classes to the hierarchy, representing 'clusters' of specific classes – for instance, tables and chairs may only appear in indoor scenes, whereas sheep and horses may only appear outdoors. The problem of learning such a visual hierarchy has been addressed in [24] (among others), and incorporating such hierarchies is certainly an avenue for future work.

These requirements will be enforced as *hard* constraints in $\mathscr{M}$ (i.e., assignments which disobey these constraints will have cost ∞). We will use the notation $a \prec b$ to indicate that a class a is *less specific* than b.

4 Probability maximization in $\mathscr{M}$

4.1 Unary potentials

In the most general case, the unary potentials (i.e., first order probabilities) in $\mathscr{M}$ are defined as follows (note that we have suppressed the subscript of the region x for brevity):

$$E(x;y) = \left\langle \Phi^1(x,y), \theta^{\text{nodes}} \right\rangle, \tag{2}$$

where $y \in \mathscr{H}$ is the class label to which the region x is to be assigned, and θ^{nodes} parametrises $\Phi^1(x,y)$ – the *joint feature map* of the node x and its assignment y. The optimal value of θ^{nodes} will be determined by our learning scheme, described in section 4.5.

In our specific case, we wish to ensure that the class label given to x was given a high probability according to some first-order classifier (such as that defined in [6]). Specifically, suppose that such a classifier returns a vector of probabilities P_x; then our joint feature map may be

$$\Phi^1(x,y) = \underbrace{(0,\ldots,P_{x,y},\ldots,0)}_{\text{0 everywhere except the } y^{\text{th}} \text{ entry}}. \tag{3}$$

In words, if x is assigned the class y, then the probability given by the first-order model should be high (weighted by θ_y^{nodes}).[3]

[3] To simplify, $E(x;y) = P_{x,y}\theta_y$. The formulation in (eq. 3) is used simply to express this as a linear function of

There are two straightforward generalisations of this model which may be desirable: firstly, we may wish to learn a separate parametrisation for each level; in this case, we would have a copy of $\Phi^1(x,y)$ for each image level, and use an indicator function to 'select' the current level. The second generalisation would be to parametrise *multiple* first-order classifiers, which return probabilities $P_x^1 \cdots P_x^C$ (for instance, features based on histograms of orientations; features based on RGB statistics, etc.). In this case, our joint feature map will simply be a concatenation of the individual feature maps defined by each classifier (i.e., it will be nonzero in exactly C locations). We will use both of these generalisations in our experiments (see Section 5).

4.2 Pairwise potentials

Similarly, we define pairwise potentials for nodes x_k and x_{k+1} (suppressing the remainder of the index):

$$E(x_k, x_{k+1}; y_k, y_{k+1}) = \left\langle \Phi^2(x_k, x_{k+1}; y_k, y_{k+1}), \theta^{\text{edges}} \right\rangle. \tag{4}$$

This time the joint feature map Φ^2 should express two properties: firstly, the constraints of our hierarchy should be strictly enforced; secondly, nodes assigned to the same class on different levels should have similar probabilities (again using the probabilities P_{x_k} and $P_{x_{k+1}}$ returned by our first-order classifier).

To achieve these goals, we define the indicator function H as

$$H(y_k, y_{k+1}) = \begin{cases} \infty & \text{if } y_k \succ y_{k+1}, \\ 0 & \text{if } y_k \prec y_{k+1}, \\ 1 & \text{otherwise } (y_k = y_{k+1}). \end{cases} \tag{5}$$

Note that this indicator function enforces precisely the hierarchical constraints that we desire. It also specifies that there is no cost associated to assigning a child node to a more specific class – thus we are only parametrising the cost when both class labels are the same. Our joint feature map now takes the form

$$\Phi^2(x_k, x_{k+1}; y_k, y_{k+1}) = -H(y_k, y_{k+1}) \left| P_{x_k} - P_{x_{k+1}} \right|^2, \tag{6}$$

where $|p|$ is the *elementwise* absolute value of p. Again we may make the same extensions to this model as outlined in Section 4.1.

4.3 The potential function

The complete maximization function is now defined as

$$g_\theta(\mathscr{X}) = \underset{\mathscr{Y}}{\operatorname{argmax}} \sum_{x \in \mathscr{M}} \left\langle (0, \ldots, P_{x,y(x)}, \ldots, 0), \theta^{\text{nodes}} \right\rangle + \sum_{x_k, x_{k+1} \in \mathscr{M}} \left\langle H(y_k(x_k), y_{k+1}(x_{k+1})) \left| P_{x_k} - P_{x_{k+1}} \right|^2, \theta^{\text{edges}} \right\rangle, \tag{7}$$

where θ is simply the concatenation of our two parameter vectors $(\theta^{\text{nodes}}; \theta^{\text{edges}})$, and $y(x)$ is the assignment given to x under $\mathscr{Y}$ (the full set of labels). As the nodes in $\mathscr{M}$ form a tree, this energy can be maximized via *max-sum belief propagation* (see, for example [□]). The running time of this procedure is in $O(|\mathscr{M}||\mathscr{H}|^2)$, where $|\mathscr{M}|$ is the number of nodes and $|\mathscr{H}|$ is the number of classes.

θ^{nodes}. Also, we use $\log(P_{x,y})$ in practice, since we are maximizing a sum rather than a product.

4.4 Loss function

Our loss function specifies 'how bad' a given assignment $\mathscr{X}$ is compared to the correct assignment $\mathscr{Y}$. We desire that our loss function should decompose as a sum over nodes and edges in $\mathscr{M}$ (for reasons shown in Section 4.5).

Firstly, we must specify how our training labels $\mathscr{Y}$ are produced using existing datasets. One option is simply to assign the class 'multiple' to all regions with which multiple bounding boxes intersect, to assign 'background' to all regions with which *no* bounding boxes intersect, and to assign a specific class label to all others. Specifically, we will define a loss function of the form

$$\Delta(\mathscr{X},\mathscr{Y}) = \sum_{i=1}^{|\mathscr{X}|} \delta(x_i, y_i). \tag{8}$$

One such loss is the *Hamming loss*, which simply takes the value 0 when the region is correctly assigned, and $1/|\mathscr{M}|$ otherwise, where $|\mathscr{M}|$ is the number of regions (or more formally $\delta(x_i,y_i) = \frac{1}{|\mathscr{M}|}(1 - I_{\{x_i\}}(y_i))$).[4] In practice, we scale the loss so that each level of our graphical model makes an equal contribution (i.e., a mistake on level k makes four times the contribution as a mistake on level $k+1$).

4.5 Structured learning in $\mathscr{M}$

Structured learning can now be done in the framework described in [29]. Given a training set $\mathscr{Y}^1 \cdots \mathscr{Y}^N$, our goal is to solve

$$\operatorname*{argmin}_{\theta} \Big[\underbrace{\frac{1}{N} \sum_{n=1}^{N} \Delta(g_\theta(\mathscr{X}^n), \mathscr{Y}^n) +}_{\text{empirical risk}} \underbrace{\lambda \|\theta\|^2}_{\text{regularisation term}} \Big]. \tag{9}$$

Without going into the details of the structured learning algorithm itself (see [29]), we require that in addition to being able to solve (eq. 7), we can also solve

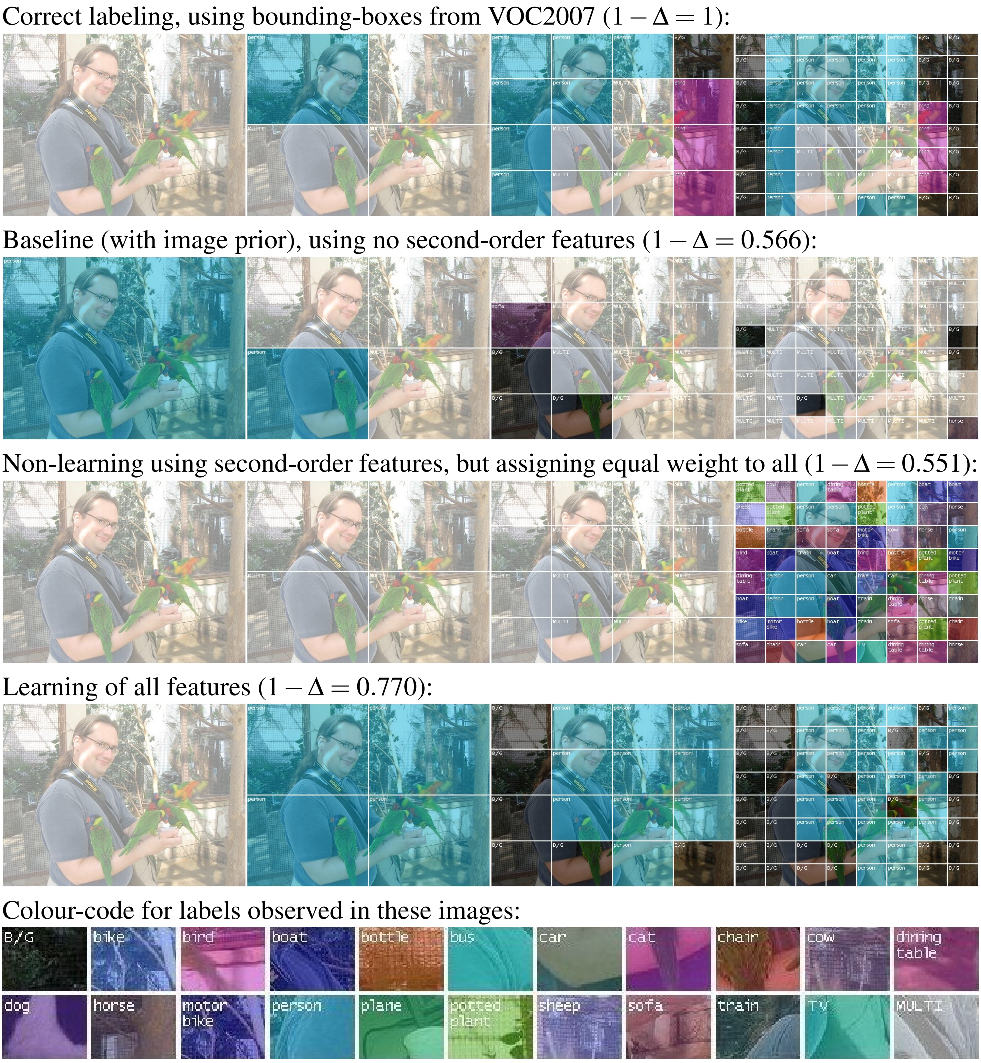
$$\begin{aligned} g_\theta(\mathscr{X}) = \operatorname*{argmax}_{\mathscr{Y}} \sum_{x \in \mathscr{M}} \left\langle (0, \ldots, P_{x,y(x)}, \ldots, 0), \theta^{\text{nodes}} \right\rangle + \\ \sum_{x_k, x_{k+1} \in \mathscr{M}} \left\langle H(y_k(x_k), y_{k+1}(x_{k+1})) \left| P_{x_k} - P_{x_{k+1}} \right|^2, \theta^{\text{edges}} \right\rangle + \Delta(\mathscr{X},\mathscr{Y}), \end{aligned} \tag{10}$$

i.e., for a given value of θ, we can find an assignment which is consistent with our model (eq. 7), yet incurs a high loss. This procedure is known as 'column-generation', and can easily be solved in this scenario, as long as $\Delta(\mathscr{X},\mathscr{Y})$ decomposes as a sum over the nodes and edges in our model (which is certainly true of the Hamming loss).

5 Experiments

We used images from the VOC2007 and VOC2008 datasets to evaluate our model. Specifically, we used VOC2008 for training and validation, and VOC2007 for testing (as testing

[4] When multiple classes are observed in a single region (i.e., when the correct label is 'multiple'), no penalty is incurred if one of these specific classes is chosen.


Correct labeling, using bounding-boxes from VOC2007 ($1-\Delta = 1$):

Baseline (with image prior), using no second-order features ($1-\Delta = 0.566$):

Non-learning using second-order features, but assigning equal weight to all ($1-\Delta = 0.551$):

Learning of all features ($1-\Delta = 0.770$):

Colour-code for labels observed in these images:

Figure 3: An example match comparing our technique to non-learning methods. The top sequence contains the 'correct' labeling, as extracted from the VOC2007 dataset (Image 000127; the correct labeling incurs zero loss by definition). The second sequence uses only first order features (i.e., the most likely assignment is chosen for each region independently); the image-level classifier is used at the top level, the mid-level classifier is used at the second and third levels, and the patch-level classifier is used at the bottom level; the fact that many of regions at the bottom level are incorrectly labeled demonstrates the need for consistency constraints (see supplementary material for further analysis). The third sequence shows our method without learning (i.e., assigning equal weight to all features); the low quality of this match demonstrates that the weights are poorly tuned without learning. Finally, the fourth sequence shows the performance of our method, which appears to address both of these issues. A key for the labels used is also shown.

	Level 0	Level 1	Level 2	Level 3
Baseline (see [6])	0.342	0.214	0.232	0.163
Baseline with image prior (see [6])	0.342	0.217	0.242	0.169
Non-learning	**0.426**	0.272	0.137	0.112
Learning	0.413	**0.307**	**0.349**	**0.444**

Table 1: Performance on each level of the graph (on the test set).

data for VOC2008 is not available). This presents a learning scenario in which there is a realistic difference between the training and test sets. The training, validation, and test sets contain 2113, 2227, and 4952 images respectively. The validation set is used to choose the optimal value of the regularisation constant λ.

We extracted SIFT-like features for our model on uniform grids at 5 different scales [19]. We used the methodology of [6] based on Fisher vectors to extract a signature for each region. A patch is considered to belong to a region if its centre belongs to that region, and its overlap with the region is at least 25%. We used three different first-order classifiers, based on sparse logistic regression: one which has been trained to classify the entire collection of features in an image (the 'image-level' classifier), one which has been trained on bounding-boxes (the 'mid-level' classifier), and one which has been trained on individual patches (the 'patch-level' classifier). The baseline to which we compare our method is one which simply selects the highest score using these individual classifiers (i.e., no consistency information is to be used). This baseline is similar to what is reported in [6], though it is important to stress that their method was not optimised to minimize the same loss that is presented here. We also report the performance using the image prior defined in [6], which rejects labelings at the patch level which are inconsistent with the probability scores at the image level.

Classification scores for the classes 'background' and 'multiple' were extracted automatically from the first-order scores: the probability of belonging to the background is 1 minus the highest probability of belonging to any other class; the probability of belonging to multiple classes is the twice the product of the two highest probabilities, capped at 1 (so that if two classes have probabilities greater than 0.5, the product will be greater than 0.5 also).

Structured learning was performed using the 'Bundle Methods for Risk Minimization' code of [28]. This solver requires only that we specify our feature representation $\Phi(\mathscr{X},\mathscr{Y})$ (eq. 3, 6), our loss $\Delta(\mathscr{X},\mathscr{Y})$ (eq. 8), and a column-generation procedure (eq. 10).

A performance comparison between the learning and non-learning versions of our approach, as well as the baseline is shown in Table 2.[5] Figure 3 shows an example match from our test set. Table 1 shows the contribution to the loss made by each level of the graphical model. Finally, Figure 4 shows the weight vector learned by our method. Note that our model exhibits a substantial improvement over the baseline, and non-learning approaches.[6]

Additional results are given in our supplementary material (including 3 × 3 branching in our tree, different weightings for the class 'multiple', and an analysis of our failure cases).

[5] The non-learning version of the approach just sets $\theta = (1,1,\ldots,1,1)$, though any constant value will do.

[6] Comparison with other methods is certainly difficult, as our loss is neither equivalent to a classification nor a segmentation error. Note however that in Table 1, we achieve an improvement at *both* the segmentation and classification levels. Also, due to the class 'multiple', our loss is *not* equivalent to the criteria normally used to measure performance on the VOC2007 and VOC2008 datasets.

	Training	Validation	Testing
Baseline (see [6])	0.272 (0.004)	0.273 (0.004)	0.233 (0.003)
Baseline with image prior (see [6])	0.275 (0.005)	0.276 (0.004)	0.238 (0.003)
Non-learning	0.235 (0.006)	0.224 (0.005)	0.233 (0.004)
Learning	**0.460** (**0.006**)	**0.456** (**0.006**)	**0.374** (**0.004**)

Table 2: Performance of our method during training, validation, and testing (for the optimal value of λ), compared to non-learning methods. The value reported is simply the proportion of correctly labeled regions, with each level contributing equally (i.e., one minus the loss). Values in parentheses indicate the standard error (across all images).

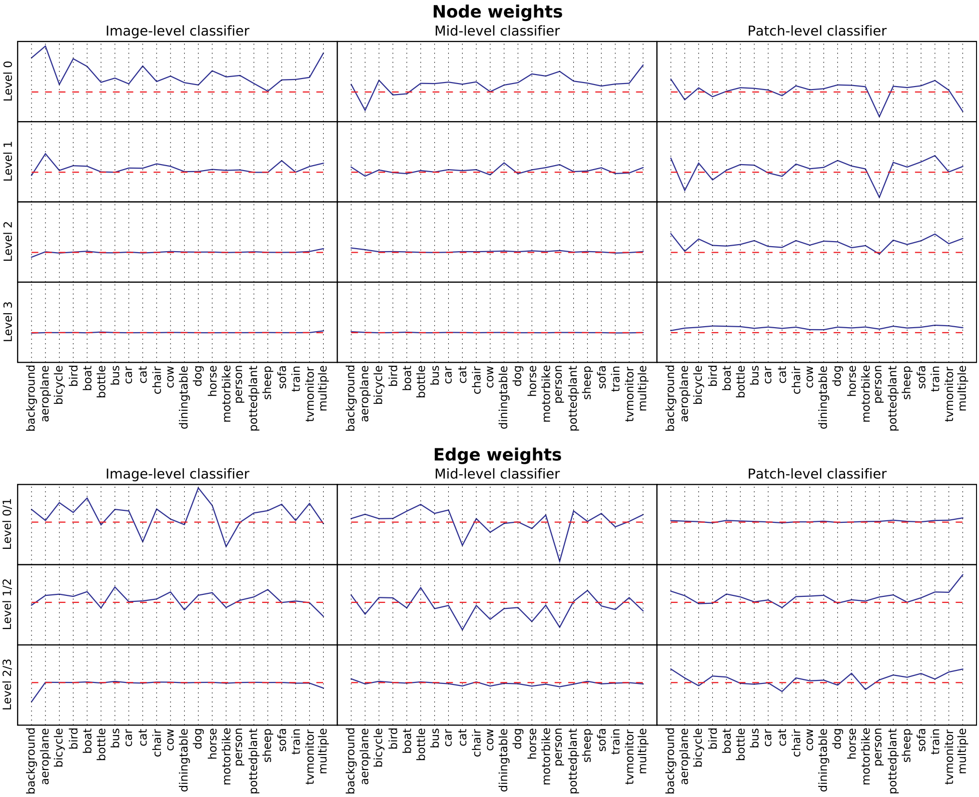


Figure 4: The complete weight-vector for our model. A separate vector of 22 ($= |\mathscr{H}|$) weights is learned for each image level, and for each type of classifier (the dashed line corresponds to zero). This vector has several interesting properties: firstly, the image-level classifier is given higher importance at the top levels, whereas the patch-level classifier is given higher importance at the bottom levels (which is consistent with our expectations). Also, there is a lot of variance between weights of different classes (especially 'multiple', 'background', and 'person'), indicating that certain classes are easier to identify at different scales. Finally, the edge weights have both large positive and negative scores: for instance, the high weight given to 'dog' for the image-level classifier at level 0/1 indicates that the features between these two levels should be very similar, whereas the *negative* weight given to 'cat' (at the same level) indicates that the features should be very *different*.

6 Conclusion

We have presented a graphical model which efficiently performs patch-level, region-level, and image-level labeling simultaneously. This model is useful in that it allows us to encode smoothness constraints for patch-level classification, while still incorporating the important information at higher levels. We have shown how to apply structured learning in this model, which has traditionally been a problem for models incorporating smoothness constraints. We have shown that our model improves in performance over existing models which use only first-order information. Since our model is parametrised using the probability scores from first-order approaches, it should be seen as complimentary to existing first-order techniques.

Acknowledgements

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n^o 216529.

References

[1] S.M. Aji and R.J. McEliece. The generalized distributive law. *IEEE Trans. on Information Theory*, 46(2):325–343, 2000.

[2] Matthew B. Blaschko and Christoph H. Lampert. Learning to localize objects with structured output regression. In *ECCV*, 2008.

[3] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. on PAMI*, 26(9):1124–1137, 2004.

[4] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. on PAMI*, 23(11):1222–1239, 2001.

[5] T. Cour, N. Gogin, and J. Shi. Learning spectral graph segmentation. In *AISTATS*, 2005.

[6] G. Csurka and F. Perronnin. A simple high performance approach to semantic segmentation. In *BMVC*, 2008.

[7] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html, 2007.

[8] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results. http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html, 2008.

[9] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008.

[10] T. Finley and T. Joachims. Training structural SVMs when exact inference is intractable. In *ICML*, 2008.

[11] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *ICCV*, 2005.

[12] X. He, R.S. Zemel, and M.Á. Carreira-Perpiñán. Multiscale conditional random fields for image labeling. *CVPR*, 2004.

[13] H. Ishikawa. Exact optimization for markov random fields with convex priors. *IEEE Trans. on PAMI*, 25(10):1333–1336, 2003.

[14] P. Kohli, A. Shekhovtsov, C. Rother, V. Kolmogorov, and P. Torr. On partial optimality in multi-label MRFs. In *ICML*, 2008.

[15] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? In *ECCV*, 2002.

[16] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

[17] V. Lempitsky, C. Rother, and A. Blake. Logcut: Efficient graph cut optimization for markov random fields. In *ICCV*, 2007.

[18] J. Li, R.M. Gray, and R.A. Olshen. Multiresolution image classification by hierarchical modeling with two-dimensional hidden markov models. *IEEE Trans. on Information Theory*, 46(5):1826–1841, 2000.

[19] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[20] D. Ramanan and D.A. Forsyth. Finding and tracking people from the bottom up. *CVPR*, 2003.

[21] D. Scharstein and C. Pal. Learning conditional random fields for stereo. *CVPR*, 2007.

[22] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. on PAMI*, 22(8): 888–905, 2000.

[23] J. Shotton, J. Winn, C. Rother, and A. Criminisi. TextonBoost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *IJCV*, 81(1):2–23, 2009.

[24] J. Sivic, B.C. Russell, A. Zisserman, W.T. Freeman, and A.A. Efros. Unsupervised discovery of visual object class hierarchies. In *CVPR*, 2008.

[25] B. Stenger, A. Thayananthan, P. H. S. Torr, and R. Cipolla. Model-based hand tracking using a hierarchical bayesian filter. *IEEE Trans. on PAMI*, 28(9):1372–1384, 2006.

[26] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *IEEE Trans. on PAMI*, 30(6):1068–1080, 2008.

[27] Martin Szummer, Pushmeet Kohli, and Derek Hoiem. Learning crfs using graph cuts. In *ECCV*, 2008.

[28] C.H. Teo, Q. Le, A.J. Smola, and S.V.N. Vishwanathan. A scalable modular convex solver for regularized risk minimization. In *KDD*, 2007.

[29] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *Predicting Structured Data*, pages 823–830, 2004.

[30] J.S. Yedidia, W.T. Freeman, and Y. Weiss. Generalized belief propagation. In *NIPS*, 2000.